Finding local RNA motifs using covariance models

Sohrab P. Shah and Anne Condon
Department of Computer Science,
University of British Columbia,
Vancouver, BC, Canada
sshah, condon@cs.ubc.ca

Technical Report # TR-2006-06

March 23, 2006

Abstract

We present DISCO, an algorithm to detect conserved motifs in sets of unaligned RNA sequences. Our algorithm uses covariance models (CM) to represent motifs. We introduce a novel approach to initialise a CM using pairwise and multiple sequence alignment. The CM is then iteratively refined. We tested our algorithm on 26 data sets derived from Rfam seed alignments of microRNA (miRNA) precursors and conserved elements in the untranslated regions of mRNAs (UTR elements). Our algorithm outperformed RNAProfile and FOLDALIGN in measures of sensitivity and positive predictive value, although the running time of RNAProfile was considerably faster. The accuracy of our algorithm was unaffected by properties of the input data and performed consistently under different settings of key parameters. The running time of DISCO is $O(N^2L^2W^2 + NL^3)$ where W is the approximate width of the motif, L is the length of the longest sequence in the input data, and N is the number of sequences. Supplemental material is available at: `http://www.cs.ubc.ca/~sshah/disco`.

1 Introduction

This work was originally intended to be submitted as a journal article, but due to the recent publication of similar work by Yao *et al.* [42], we have elected to publish this work as a technical report.

Non-coding RNAs (ncRNAs) are functional RNAs which are transcribed from DNA, but do not get translated into proteins. Dubbed "the architects of eukaryotic complexity" [33], ncRNAs play diverse and essential roles in the cellular machinery of eukaryotes that extend well beyond the traditional central dogma of transcription and translation [8]. Examples of ncRNAs include microRNAs (miRNAs), which are small, approximately 22-nucleotide, molecules that bind to mRNAs and influence protein expression [24, 26]. Other examples are RNA elements, which, unlike ncRNAs, are embedded in other RNA molecules. RNA elements in the untranslated regions (UTRs) of mRNA transcripts of certain genes are essential for regulation of translation [6, 14, 15, 25]. It is of great interest to detect such ncRNAs computationally, when analysing a genome sequence, as a means to automatically detect candidate functional ncRNAs.

However, computational prediction of ncRNAs in genomic sequences has proved to be a significantly more difficult challenge than predicting genes that encode proteins, due to a lack of obvious statistical signals emitted by ncRNAs in genomic sequences [36]. Additional signal can be obtained by considering secondary structure constraints that are maintained through evolution. Such signals can be further leveraged by considering a set of homologous RNAs which have evolved under the same constraints as the ncRNA of interest. In this paper, we describe a new algorithm for prediction of *RNA motifs* – functional RNAs contained *locally* in each sequence of a set of *unaligned*, homologous, sequences. We use the term *RNA motif* to connote a pattern, shared by a set of (sub)sequences, which incorporates both sequence and secondary structure. By local, we mean that the functional RNA may not span the length of the sequences; the case when the functional RNA does span the length of the sequences is the global version of the problem.

Our work builds on a large body of earlier work that addresses related problems. Eddy and Durbin [10] introduced the covariance model (CM) for specifying RNA sequences and secondary structures probabilistically. A generalisation of hidden markov model (HMM) profiles for modeling sequence motifs, a CM is a special type of stochastic context free grammar (SCFG), and can be viewed as a probabilistic generator of a family of RNA sequences and a secondary structure shared by the sequences. Compared with deterministic models such as the RNAMotif description language [31], CMs are better suited for specification of naturally occurring sequence and structure patterns which have emerged as the result of the stochastic process of evolution.

The Rsearch algorithm [21] (built on an earlier Infernal algorithm of Eddy [9]), provides a local alignment tool that can be used to search a database of sequences for homologues of a single RNA sequence with known secondary structure, represented as a CM. A key contribution of this work is a set of empirically derived substitution rates (called RIBOSUM matrices) for unpaired nucleotides and base-paired nucleotides. RIBOSUM matrices can be used to construct a CM from a single sequence with known secondary structure. In addition this work contributes a local alignment variant of Eddy and Durbin's CM global alignment algorithm [10, 7]. This is relevant to the RNA motif prediction problem and will be discussed further in the Methods section.

A number of recently developed algorithms can be used to predict global ncRNAs from aligned RNA sequences [17, 22, 23, 37, 40]. All of these algorithms rely on having an alignment of the RNA sequences as input, and do not change the given alignment. In particular, Alifold [17], which we use in our approach, predicts a global consensus secondary structure from a set of aligned sequences, using free energy minimization. Other work uses characteristics of specific RNA families for better predictions [29, 30]. These algorithms are, by design, highly specific to particular gene families and therefore do not generalise well for discovering unknown functional RNAs, which is a goal of the RNA motif prediction problem.

The work on global structure prediction does not apply to the *local* case, when only parts of related RNA sequences have conserved functional elements. Furthermore, of the work referenced above, only that of Eddy and Durbin [10] addresses the problem of predicting functional RNAs from *unaligned* sequences, but this solves the global case only. A good alignment can be difficult to obtain, particularly if the functional RNA does not span the entire length of the sequences. In addition, some of the work assumes that a model describing the functional RNA is already known, and thus cannot be used for finding unknown functional RNAs.

Several approaches have recently been proposed for RNA motif prediction, that is, prediction of functional RNAs contained locally in each sequence of a set of unaligned, homologous, sequences [11, 16, 34]. We focus our discussion below on two of these, which are most relevant for comparison with our approach: the SLASH/FOLDALIGN algorithm of Gorodkin *et al.* [12], and the RNAProfile algorithm of Pavesi et al. [34]. In contrast, other approaches either produce an output in a format that is not compatible with CMs [16, 39], require rather specific information about the motif from the user [19], or are not strongly guided by thermodynamic criteria [20]. However, the approach of Ji *et al.* [20] is notable in several respects: it can predict pseudoknotted motifs not shared by all sequences, and it reports

a given number of best scoring motifs. These authors did a thorough comparison of their approach with other published methods and present a quantitative measure for evaluation of performance, similar to that used in our paper.

The FOLDALIGN algorithm of Gorodkin *et al.* for the RNA motif prediction problem is based on an extensive alignment approach. All pairs of input sequence are aligned, using a 4-D dynamic programming algorithm that takes both structure and sequence similarity into account. Then, a greedy strategy is used to build multiple alignments from the pairwise alignments. The SLASH algorithm of Gorodkin *et al.* [12] uses FOLDALIGN to generate a seed alignment and consensus secondary structure using just a subset of the input sequences, from which a CM is initialized. The remaining sequences are then aligned to the CM, and the result is output as a CM. SLASH is faster than FOLDALIGN, but still needs $\emptyset(N^4L^4)$ time to create the initial CM, where N is the number of sequences used to create the seed alignment and L is the length of the longest sequence, making it feasible only on small inputs.

The RNAProfile algorithm of Pavesi et al. [34] outputs aligned regions of a set of unaligned RNA sequences and a consensus structure for these sequences, where the alignment has a high similarity score that accounts for both sequence and secondary structure. For this algorithm, the input includes the number of stems that should be in the functional RNA, in addition to the unaligned sequences. The algorithm first selects from the input sequences a set of candidate subsequences (regions), each of whose secondary structures (derived by dynamic programming with a thermodynamic energy model) contains a given number of stems for the subsequence. High-scoring profiles – groups of aligned candidates – are then built in a greedy fashion. The algorithm, which does not represent RNA motifs as CMs, has a running-time of $\emptyset(N^2L^2)$ – much faster in practice than the FOLDALIGN/SLASH algorithm. The use of profiles enables the algorithm to detect instances of the motif that may have diverged considerably.

In this paper, we present DISCO, a new algorithm for RNA motif prediction. We assume, for efficiency reasons, that an approximation to the size (length) of the motif is known. Our goal is to improve on the quality of motif detection obtained by RNAprofile and FOLDALIGN, within a reasonable running time. We measure the quality of a CM using a sum of log-odds scores for the CM with respect to each input sequence. Additionally, we use sensitivity and positive predictive value to measure the accuracy of our predictions.

DISCO has an initialization phase, which improves on the complexity of SLASH for initialising a CM, followed by an iterative refinement phase to improve the initial CM. For efficiency reasons, the initialization phase uses two heuristics: the CM is initialized using a small subset of the input sequences, and a filter further removes from consideration subsequences of this subset which have low structural signal.

We use iterative refinement to improve on the initial CM, because of its success in solving related sequence motif finding problems [2, 3, 5, 27, 28]. The iterative refinement process is not guaranteed to converge on global optima, so we do not expect to always detect the highest-scoring CM. One goal of this work is to test whether, given a good starting point, iterative refinement will converge on a model that is composed of a majority of the instances of the motif. DISCO outputs a multiple sequence alignment, a consensus structure and

the score of the corresponding CM, which is compatible with Infernal and Rfam; thus it is straightforward to search a genomic sequence for good matches with the output of our algorithm.

The worst case time complexity of DISCO is $O(N^2L^2W^2 + NL^3)$, where N is the number of input sequences, W is the user-inputted approximate width of the motif and L is the length of the longest sequence in the input data.

We use seed (curated) alignments from the Rfam database to test our algorithm, with sequences flanking the ncRNAs or RNA elements obtained from GenBank/EMBL. Our data sets contain both miRNAs and UTR elements, selected in light of their important role in post-transcriptional gene regulation, and because of their manageable size (30-100 bp). We measure the quality of the motif output by our algorithm in several ways, including measures of sensitivity and positive predictive value of the degree of overlap between base pairs of the true motif and the motif discovered by our algorithm; these measures are described in detail in the Methods Section.

A comparison between RNAProfile, FOLDALIGN and DISCO shows that DISCO is more sensitive and has higher positive predictive value than both RNAProfile and FOLDALIGN. This type of comparison has not been performed in other papers describing RNA motif detection algorithms and we view our methodology for such a comparison as a contribution in itself.

Our algorithm detected the motifs in the majority of the data sets. For the miRNA data, the mean sensitivity and positive predictive value were 0.6 and 0.73 respectively – higher than those obtained by FOLDALIGN or RNAProfile. For the UTR data, the algorithm found four motifs with sensitivity of at least 0.95 and one with sensitivity 0.57, but completely missed the remaining four motifs.

Further analysis shed more insight on the efficacy of different components of our algorithm. We found that initial sequence-based alignment of subsequences was more effective overall than structure-based alignment. We also found that creation of our initial covariance model based on subsequences of six, rather than all, input sequences yielded good results, as did the elimination of subsequences that have a high proportion of unpaired bases. Finally, iterative refinement was effective. In nearly all test cases, iterative refinement either improved or maintained the accuracy of the resultant model.

2 Materials and Methods

The DISCO algorithm for RNA motif prediction takes as input a set of N unaligned RNA sequences (in FASTA format) and an approximate motif width W. The algorithm outputs a motif represented as a multiple sequence alignment of one subsequence per input sequence along with a consensus secondary structure. The width of the alignment is approximately W. The index of the parent sequence (by location in the input data) of each sub-sequence and its position in its parent sequence are also given in the output. A sample output is shown in Figure 1 of the Supplementary Material.

We describe the DISCO algorithm in two phases, the *initialisation* phase which con-

structs an initial CM, and the *iterative refinement* phase, which improves on the initial CM. Pseudocode is given in the Supplementary Material.

2.1 Initialisation phase

2.1.1 Sliding window secondary structure prediction

We enumerate windows (i.e. subsequences) of width W in the input data and predict the secondary structure of each window using an implementation of Zuker and Steigler's algorithm (see Andronescu *et al.* [1]). This step gives us a dot-bracket representation of the minimum free energy secondary structure of each W-mer in the input. In this representation, each position in the W-mer is assigned a character from the set $\{(, \cdot,)\}$. Matched '(' and ')' indicate base-paired positions and '$\cdot$' indicates unpaired positions. An overlap parameter o specifies the degree of overlap between successive windows. For example, if $W = 10$ and $o = 9$, the window slides one position and all W-mers in the input are enumerated. However, if $W = 10$ and $o = 5$ the sliding window steps skips over five positions before enumerating the next W-mer. See Algorithm 2 of the Supplementary Material for details.

2.1.2 Pairwise alignment of W-mers

Next, every pair of W-mers (W_1, W_2) enumerated in the previous step, where W_1 and W_2 are from different input sequences, are aligned (see Algorithm 2, line 12 , in the Supplementary Material). The alignment is done using the Needleman-Wunsch optimal alignment algorithm, in one of three ways:

- **Sequence**: using sequence information only with a RIBOSUM85-60 [21] scoring matrix (see Figure 2 of the Supplementary Material).
- **Structure**: using the dot-bracket representation of the secondary structure only, with a scoring matrix (called DISCOSUB) that is similar to that used by Pavesi *et al.* [34] (see Figure 3 of the Supplementary Material).
- **Combination**: using a combination of sequence and structure that uses RIBOSUM85-60 for unpaired nucleotides that align, and DISCOSUB for paired nucleotides that align.

The algorithm was implemented in this way in order to test the properties of the input data (sequence, structure or combination) that contain the strongest signals for CM initialisation. The entries in the DISCOSUB matrix were chosen (somewhat arbitrarily) in order to score aligning left '(' or aligning right ')' parentheses most highly, to score aligning dots less highly, and to penalize mismatches.

To reduce the number of pairwise alignments, we use a filter to reduce computational effort while maintaining accuracy. For each W-mer, we calculate its 'dot-composition' (DC), meaning the number unpaired nucleotides in its secondary structure divided by W. We ignore all W-mers with a DC of greater than some dot-composition threshold d. The lower d is, the fewer W-mers will be considered for pairwise alignment.

The remaining W-mers are called anchors. Furthermore, we do not align any two W-mers if their DC differ by more than 0.20 (arbitrarily selected). The k highest-scoring W-mers that align to each anchor W-mer w are stored in sorted order according to alignment score in an array $H_w = H_w[1, \ldots, k]$ with $H_w[1] = w$.

2.1.3 Multiple alignment of a set of W-mers

Each array H_w is converted to a multiple alignment using a progressive alignment technique (see Algorithm 3 , Supplementary Material). First, the alignment of w to $H_w[2]$ is converted to an alignment profile P. Each column j of P is represented by q-dimensional vector P_j containing the frequency of occurrences of each symbol from the alphabet $\{A, C, G, U, -\}$ or $\{(, \cdot,), -\}$ at position j in the alignment, where $-$ represents a gap in the alignment and q is the number of characters in the alphabet (either 5 or 4). Then P is aligned to $H_w[3]$, using dynamic programming, and a score is calculated for this alignment. Note that if (gapped) sequence w' is aligned with profile P, the score S_{ij} for aligning position i of w' to position j of P is $\sum_\alpha P_{j_\alpha} M_{\alpha w'_i}$ where P_{j_α} is the frequency of character α in column j of P, w'_i is the character at position i of w', and M is the scoring matrix (one of RIBOSUM85-60 or DISCOSUB). P is then updated to be the profile of an alignment of $H_w[1](= w)$, $H_w[2]$ and $H_w[3]$. P is iteratively aligned and updated until all W-mers in H have been aligned to P. At the end of this step, we have a multiple alignment A of the k W-mers with highest-scoring pairwise alignments to w. We store a fixed number l of the highest scoring multiple alignments, where the score is determined by the profile alignment algorithm outlined in Durbin *et al* [7]. These are then passed to the refinement phase.

2.1.4 Prediction of consensus structure from multiple alignment

For each of the l models constructed in in the previous step, a consensus structure is predicted using Alifold [17]. A CM for each model consensus secondary structure is then created using the `cmbuild` routine from the Infernal package [9].

2.2 Iterative refinement phase

Using the l initialised CMs, we apply the INSIDE alignment algorithm `cmsearch` of the Infernal package [9] to align each CM to each sequence in the input (see Supplementary Material, Algorithm 5, line 9). Using the gapped representation of the best scoring 'hit' for each sequence, a new multiple alignment is created. We score the resultant multiple alignment, and thus the CM, as the sum of the bit scores for each hit. The bit score is a log-odds score that is the difference of the likelihood of the hit aligning to the CM (calculated by the INSIDE algorithm) and the likelihood of a random sequence aligning to the CM. As in the initialisation phase, a consensus secondary structure from this new alignment is then predicted with Alifold and a new, refined, CM is built from the multiple sequence alignment and consensus secondary structure. The refined CM is realigned to the sequences to generate a new multiple alignment and a new consensus secondary structure. This process is repeated until the score of the multiple alignment no longer improves, or a

maximum number T of iterations is reached. The algorithm outputs the highest scoring CM detected in the refinement phase.

2.3 Implementation

The algorithm is implemented in the C/C++ programming language. All functions are implemented in C, but the main executable file is implemented in C++ due to a dependency on a C++ library. The DISCO implementation is dependent on Infernal version 0.55 and the Vienna package version 1.5. A distribution package for DISCO is available at http://www.cs.ubc.ca/ sshah/software/disco/disco-0.1.tar.gz. While we have only compiled our code under Linux, we expect the code to compile under most Unix platforms.

2.4 Parameters

Parameters used by the algorithm are of three types (summarized in Table 1 of the Supplementary Material):

- **Required parameters**: A key parameter is W, the width of the motif. Another key parameter is a, the method of sequence alignment. If users expect a strong secondary structure signal, they can choose the 'structure' method, or they can choose the 'sequence' method if they expect the motif to be highly conserved at the sequence level.

- **Running-time parameters**: There are several running-time enhancing parameters, described in earlier sections: the number o of overlapping nucleotides in W-mers in the initialization step, the dot-composition threshold d, the number k of subsequences in each multiple alignment, the number l of models on which to run the refinement phase, and the number T of refinement iterations.

- **Matrix parameters**: Other matrices can be used in place of RIBOSUM85-60 or DISCOSUB. They must be in the same format as depicted in Figure 2 and Figure 3 of the Supplementary Material.

2.5 Complexity

The worst case time complexity of DISCO is $O(N^2L^2W^2 + NL^3)$, where N is the number of input sequences, W is the user-inputted approximate width of the motif and L is the length of the longest sequence in the input data. The time needed for sliding window secondary structure prediction (section 2.1.1) is $O(NLW^3)$, since in the worst case the number of W-mers is NL, and the time to fold each one is $O(W^3)$. This is dominated by the $O(N^2L^2W^2)$ term in our expression for time complexity. The $O(N^2L^2W^2)$ term accounts for pairwise alignment of W-mers (section 2.1.2). The maximum number of pairwise alignments is $(NL)^2$, and each alignment takes $O(W^2)$ time. Finally, the $O(NL^3)$ term accounts for the refinement

phase (section 2.2) in which the INSIDE algorithm (the `cmsearch` method), which has $O(L^3)$ running time, is run on each of the N input sequences.

In obtaining the above bound, we assume that the parameters k (the number of sequences that inform each model, see section 2.1.3), l (the number of models, see section 2.1.3, and T (the number of refinement iterations, see section 2.2) are constants. In practice, the time for the sliding window secondary structure prediction can be significantly reduced by setting the overlap parameter o to be low, and the time needed for pairwise alignment of W-mers can be significantly reduced by setting the dot-composition threshold d to be low, with some cost in accuracy.

2.6 Data

We constructed two data sets: a training set, used to determine how to set parameters, and a test set, used to evaluate the performance of our algorithm once the parameters were fixed. The training set consisted of eight arbitrarily chosen Rfam seed alignments (six miRNAs, namely RF00047, RF00104, RF00129, RF00237, RF00241, and RF00256 and two UTRs, namely RF00172 and RF00180).

For our test data, we selected seventeen miRNA families and nine UTR element families from the Rfam database using the keyword searches 'microRNA' and 'UTR' on the Rfam website (http://www.sanger.ac.uk/Software/Rfam/). MicroRNAs and UTR elements were selected in light of their important role in post-transcriptional gene regulation and they were of ideal size (30-100 bp) for use in prototyping our algorithm.

We used the Rfam seed alignments as 'ground truth' alignments for testing the DISCO algorithm. The seed alignments are curated multiple alignments of individual members of an RNA family. The consensus secondary structure is annotated on this multiple alignment. Rfam uses these seed alignments and secondary structures to construct CMs, which are then used to search large genomic databases for other members of the family.

An example Rfam seed alignment in Stockholm format (see Eddy [9]) is given in Figure 4 of the Supplementary Material.

From the initial set of families retrieved with the keyword searches, we removed all families with fewer than four members, with more than twenty members, with length more than 151 and UTR element families whose members extended into coding sequence. The last criterion reflects our opinion that protein coding sequences have distinct properties that would confound their analysis. We did not impose any taxonomic filters. Tables 2 and 3 of the Supplementary Material list and describe some characteristics of the nine UTR data sets and seventeen miRNA data sets used in this analysis. Using the larger 'parent' sequences given by the GenBank accession numbers in the seed alignments, we constructed the test data sets as follows: for UTR data, the entire UTR in which the seed sequence was embedded was extracted; for miRNA data, the miRNA plus 200 nucleotides upstream and downstream of the miRNA were extracted. In some cases, extracting 200 nucleotides was not possible due the proximity of the miRNA to an end of the sequence. In such cases, we extracted as much flanking sequence as possible to the end of the sequence.

2.7 Evaluation method

We evaluated the quality of the output produced by DISCO in several ways.

2.7.1 CM score

First, as mentioned above, the algorithm outputs a score, which reflects the quality of the CM model output by the algorithm. The score, which we denote by SC, is a sum of the likelihood of the model given each sequence. The higher SC is, the better the alignment. We assessed the correlation of the score SC to the measures listed below to determine whether a higher score meant better performance.

While SC measures the quality of the alignment, but this is an insufficient measure on its own, since the algorithm may produce a very high scoring alignment that does not contain members of the Rfam seed alignment. This could arise if the input data contained other regions of similarity that were more easily detectable than the members of the Rfam seed alignment. Therefore, we also use additional metrics.

2.7.2 Degree of nucleotide-level overlap

A useful measure is the number of nucleotides from each sequence included both in the output and in the Rfam seed alignment; we call these overlapping nucleotides.

To get a quantitative measure of the degree to which nucleotides overlap, we chose to use sensitivity (NS) and positive predictive value (NPPV). To define NS and NPPV, we first need to describe three other terms:

- true positives (TP): the number of pairs of nucleotides in the output that were also in the Rfam seed alignment
- false positives (FP): the number of pairs of nucleotides in the output that were not in the Rfam seed alignment
- false negatives (FN): the number of pairs of nucleotides not in the output that were in the Rfam seed alignment

NS is defined as TP/(TP+FN), or the number of true positives over the total number of aligned nucleotides in the Rfam seed alignment. NPPV is defined as TP/(TP+FP), or the number of true positives over the total number of aligned nucleotides in the output. Often, when measuring accuracy in tests of this nature, specificity – defined as TN/(TN+FP) – is used as a complementary measure to sensitivity. However, it is not clear how to define TN in our context, so we use positive predictive value instead.

Using NPPV and NS, we can now approximate the Matthews correlation coefficient [32] (NCC) in the following way.

$$NCC \approx \sqrt{NS \cdot NPPV} \tag{1}$$

This measure was originally used by Gorodkin *et al.* [12] and appears in several subsequent papers [19, 20, 13]. Since we are interested in modeling and recovering motifs, our measures

are slightly different and focus on the aligned nucleotides and recovering specific nucleotides that are part of the motif. This is another advantage of using the Rfam seed alignments in that we have a nucleotide-level 'ground-truth' with which to compare our results.

2.7.3 Degree of sequence-level overlap

We also define accuracy at a coarse level which gives a measure of whether the motif was found in each sequence or not. We define a TP in this scenario if 50% of the overlapping nucleotides in each sequence are TP nucleotides. FP and FN can be similarly defined. From these, we can define sensitivity, specificity, and correlation coefficient, which we denote by SS, SPPV and SCC for sequence-level overlap.

3 Results

3.1 Preliminary experiments and parameter estimation

Preliminary experiments were run on the set of eight Rfam seed alignments in our training set, to inform our choice of three different parameters:

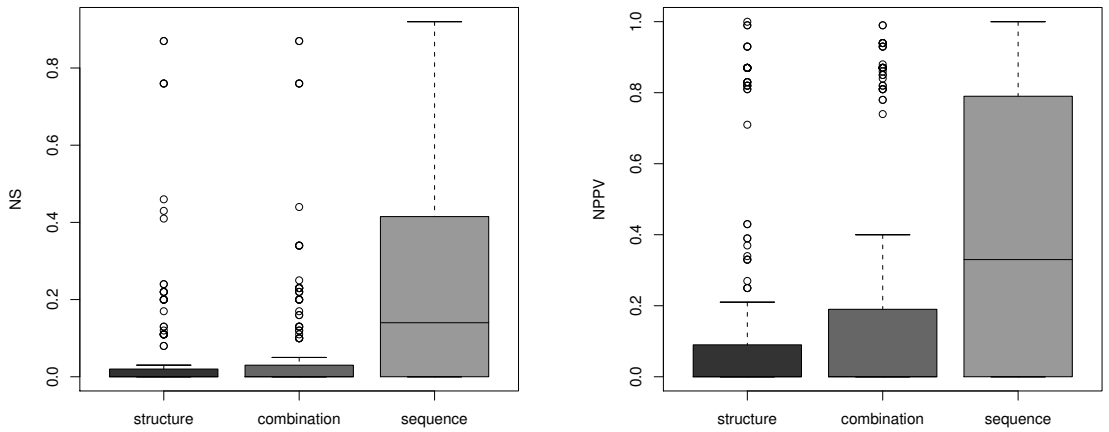
- **Alignment method a**, which is used to align subsequences in the initialization step (a being either structure alone, sequence alone, or combination): Our goal was to understand whether sequence or structure more strongly identifies a motif embedded in a set of unaligned sequences.
- **Number of subsequences k** in each multiple alignment ($k \in \{2, 3, 4, 5, 6, 7\}$): Here, our goal was to understand whether using a subset of the sequences is good enough to obtain a high-quality CM initialization.
- **Dot-composition threshold d** ($d \in \{0.45, 0.50, 0.55, 0.60, 0.65\}$): Our goal was to understand whether a filter, which eliminates from consideration those subsequences with a high number of unpaired bases in their secondary structure, would still successfully find the motif.

We did a total of 720 ($= 8 \times 3 \times 6 \times 5$) runs of the algorithm in the preliminary experiments, one for each choice of seed alignment (8 choices in all) and of the parameters a (3 choices), k (6 choices), and d (5 choices). For all experiments we chose W to be the width of the seed alignment for the input data data. We discuss the effect of choosing a more general value for W in the Discussion section. The o parameter was set to $W - 1$ for all experiments, T was set to 10 iterations and l was set to 15.

3.1.1 Choice of alignment method a

Figure 1 shows the distributions of NS and NPPV taken over all the runs, broken down by alignment method (structure, combination and sequence). All distributions in this paper are shown as box-and-whisker plots: the line within the box indicates the median of the

distribution, the top and bottom edges of the box indicate the third and first quartiles, the ends of the whiskers indicate the 95% confidence intervals of the distribution. The points shown on the plots are outside the 95% confidence intervals. The sequence alignment method overall produced higher values of NS and NPPV than did either the structure only or comparison methods. Based on these results we fixed the parameter a to be the sequence alignment method.



(a) (b)

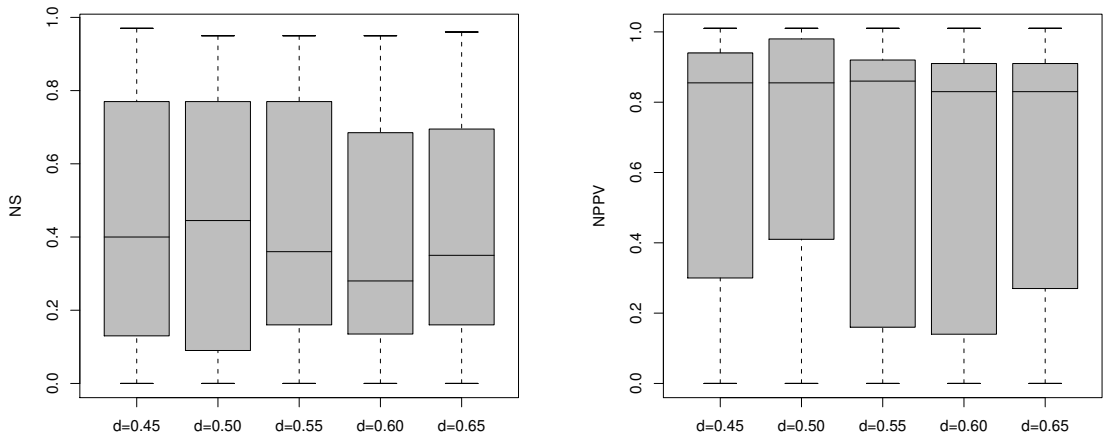
Figure 1: Box-and-whisker plots showing distribution of (a) NS and (b) NPPV for each alignment method. Distributions are taken over all 720 preliminary runs, broken down by alignment method (structure, combination and sequence). In both (a) and (b), for the structure and combination methods, the median line coincides with the bottom edge of the box. For NS, the sequence method showed significantly better performance than the structure method (Welch Two Sample t-test, t=12.84 and p=2.2E-16) and the combination method (Welch Two Sample t-test, t=12.44 and p=2.2E-16). Also, for NPPV, the sequence method showed significantly better performance than the structure method (Welch Two Sample t-test, t=13.33 and p=2.2E-16) and the combination method (Welch Two Sample t-test, t=11.35 and p=2.2E-16).

3.1.2 Choice of number of subsequences k

We next plotted NS and NPPV for our runs, broken down by values of the number of aligned subsequences, k. No value of k emerged as significantly better than the others. However, the values $k = 5, 6, 7$ produced the most accurate results. Of these values, the value $k = 6$ had the highest mean and smallest standard deviation for both NS and NPPV (data not shown). Therefore, we fixed the parameter k to be 6. For input data sets with fewer than six sequences, we set k to be N, the number of sequences.

3.1.3 Choice of dot-composition threshold d

Our goal was to choose d to be as low as possible, in order to ensure reasonable running-time, while not excluding subsequences that overlap with motifs. Figure 2 shows the distributions of NS and NPPV, for the preliminary runs with a =sequence, broken down by value of dot composition threshold d. Again, these results did not suggest a clear choice for d. Therefore, we plotted the dot composition (that is, fraction of unpaired bases) in 73 miRNA and UTR element seed alignments obtained from Rfam using keyword searches 'microRNA' and 'UTR' (see Figure 3). The mean ($\pm$ standard deviation) of the dot-composition was 0.48 ± 0.14 and the third quartile was at 0.54. Thus, we chose $d = 0.55$ to ensure that we include all subsequences corresponding to motifs whose dot composition threshold is within the third quartile of the distribution.



(a) (b)

Figure 2: Box-and-whisker plots showing distributions of (a) NS and (b) NPPV, taken over all 240 preliminary runs with a =sequence, for each value of d in the set $\{0.45, 0.50, 0.55, 0.60, 0.65\}$. The (mean, median) values for NS were (0.45, 0.40), (0.46, 0.45), (0.44, 0.36), (0.38, 0.28) and (0.42, 0.35) for successive (increasing) values of d. $d = 0.50$ had the highest values for NS, although the distribution $d = 0.50$ was not statistically significantly different from the distribution of $d = 0.45$ (Welch Two Sample t-test, t=-0.12, p=0.90) or from the distribution of $d = 0.55$ (Welch Two Sample t-test, t=-0.28, p=0.78). The mean and median values of NPPV were (0.65, 0.86), (0.67, 0.86), (0.65, 0.86), (0.61, 0.83) and (0.63, 0.83).

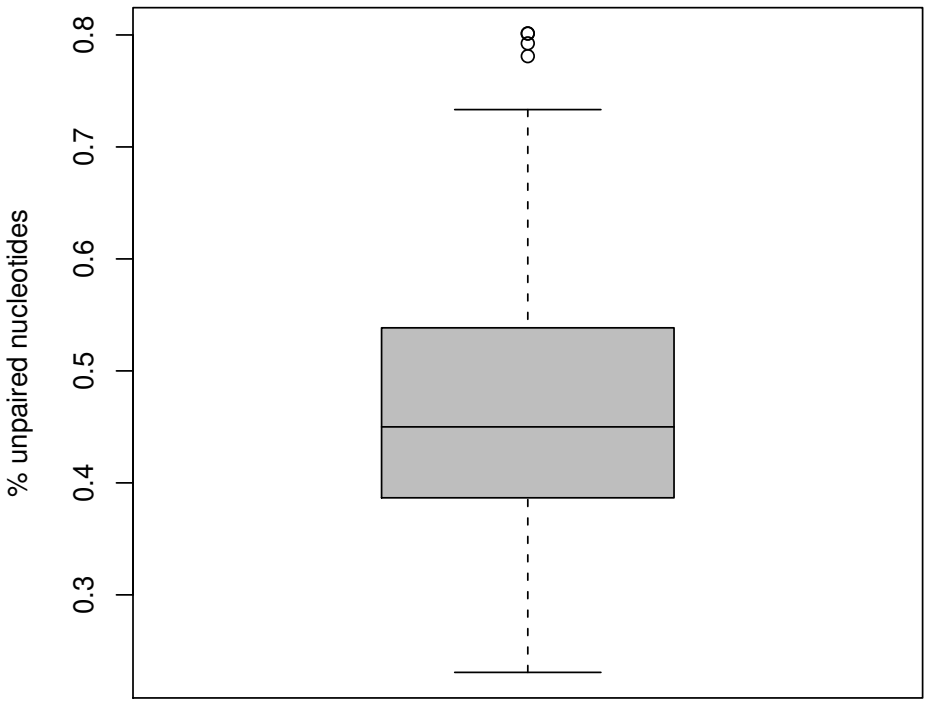


Figure 3: Box-and-whisker plot showing distribution of the proportion of unpaired nucleotides in the consensus secondary structure of 73 miRNA and UTR element seed alignments from Rfam.

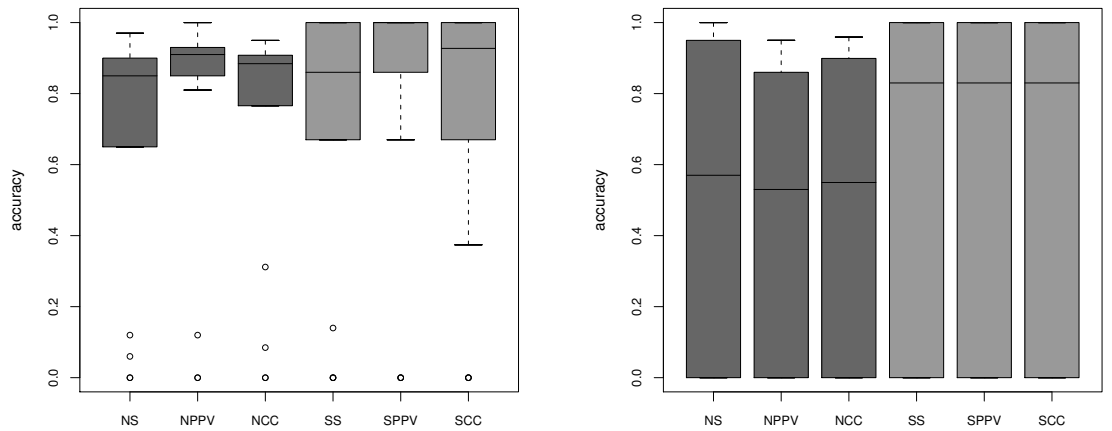
3.2 Fixed-parameter experiments

Figure 4 shows the distribution of score (SC) and accuracy measures (NS, NPPV, NCC, SS, SPPV and SCC) for the test data sets (seventeen miRNA data sets and nine UTR sets). See Tables 4 and 5 of the Supplementary Material for more details, which show score (SC) and accuracy measures (NS, NPPV, NCC, SS, SPPV and SCC) for the seventeen miRNA data sets and nine UTR data sets, respectively.

The DISCO algorithm detected the motifs for the majority of the data sets. For miRNA data, the mean and median NS were 0.60 ± 0.30 and 0.73 while the mean and median NPPV were 0.79 ± 0.32 and 0.91. Thus, on average, 60% of nucleotides in the seed alignments were found in the best scoring CM and 79% of the nucleotides found in the best CM were part

of the seed alignment. DISCO recovered at least 67% of the seed sequences in thirteen out of seventeen miRNA data sets. The mean and median SS were 0.69 ± 0.34 and 0.83 and the mean and median SPPV were 0.87 ± 0.33 and 1.00 respectively for the miRNA data. The large sources of error are most likely attributed to three of the data sets in which the motifs were essentially missed by the algorithm.

The results for the UTR element data sets were less promising. The NS mean and median were 0.49 ± 0.48 and 0.57. The NPPV mean and median were 0.45 ± 0.44 and 0.53. The performance was similarly poor by the other measures. The mean and median for both SS and SPPV were 0.53 ± 0.51 and 0.83. This relatively poor performance and very large standard deviations are due the the fact that the algorithm completely missed the motif in four of the nine UTR element data sets (see Table 2 of the Supplementary Material). The NS mean of the remaining five data sets was 0.89 ± 0.18.



(a) (b)

Figure 4: Distribution of accuracy results for (a) seventeen miRNA and (b) nine UTR element test data sets. Accuracy measures NS, NPPV, NCC, SS, SPPV and SCC are reported.

3.2.1 Score indicates sensitivity

Recall that the score, SC, of the CM is the sum of the bit scores of the best hit of each sequence aligned to the CM with the INSIDE algorithm. Of great interest to us was whether the score was a good indicator of accuracy. To test this, we first normalised SC by the number N of sequences in the input to give: $\mathrm{SC}' = SC/N$. Normalisation was necessary since the score of a high scoring CM is expected to have a 'bit score' contribution from most of the sequences in the input. We then plotted SC' against NS, NPPV, SS and SPPV for all

data in the test set, and tested each measure of accuracy for a statistical correlation with normalized score SC′ using a Pearson's product moment correlation test.

All measures were positively correlated with score and NS and SS were statistically significantly correlated (p=0.003 and p=0.008, respectively). Scatter plots of correlation against the accuracy measures along with the correlation coefficient and p-values of the correlation tests are shown in Figure 5 of the Supplementary Material. These results indicate that score is a positive indicator of sensitivity. We investigated this data further by eliminating all score-accuracy measure pairs with zero values for the accuracy measures and replotting the data. The correlation was still significant (p=0.015) for NS, however it was insignificant for SS (p=0.051).

3.3 DISCO is more accurate than RNAProfile and FOLDALIGN

Figure 5 shows the distribution over the 26 test data sets of NS, NPPV, SS and SPPV accuracy measures of the highest-scoring CM models (using the SC score) produced by DISCO, against the highest-scoring models produced by RNAProfile version 2.1 and FOLDALIGN version 0.02 (according to their scoring measures).

DISCO outperformed RNAProfile version 2.1 and FOLDALIGN version 0.02 for all accuracy measures. Mean NS was 0.57 ± 0.39 with median 0.69 for DISCO, 0.36 ± 0.25 with median 0.45 for RNAProfile and 0.47 ± 0.28 with median 0.47 for FOLDALIGN. Mean NPPV was 0.67 ± 0.39 with median 0.85 for DISCO, 0.37 ± 0.26 with median 0.45 for RNAProfile and 0.56 ± 0.43 with median 0.80 for FOLDALIGN. Also, mean SS was 0.64 ± 0.40 with median 0.83 for DISCO, 0.39 ± 0.26 with median 0.47 for RNAProfile and 0.53 ± 0.46 with median 0.82 for FOLDALIGN. Mean SPPV was 0.75 ± 0.42 with median 1.00 for DISCO, 0.39 ± 0.26 with median 0.47 for RNAProfile and 0.53 ± 0.46 with median 0.82 for FOLDALIGN.

While DISCO had better accuracy overall, it completely missed the motif for two miRNA and four UTR structures. On two structures, DISCO was highly accurate while RNAProfile and FOLDALIGN completely missed the motif. Notably, on one structure, RNAProfile was accurate, but both DISCO and FOLDALIGN missed.

Figure 6 (a) shows the running time of DISCO, RNAProfile and FOLDALIGN plotted against the size of the input data. RNAProfile had considerably faster running time than DISCO and FOLDALIGN for all data sets by approximately one order of magnitude (mean log ratio of DISCO running time to RNAProfile was 1.6 and mean log ratio of FOLDALIGN to RNAProfile was 1.5). The mean log ratio of DISCO to FOLDALIGN was 0.1 indicating that on average, FOLDALIGN ran marginally faster than DISCO. The minimum running time over all 26 sets for DISCO was 123 seconds for a data set of size 4001bp. The maximum running time was 55856 seconds (15.5 hrs) for a data of size 11783 bp. Mean running time for DISCO was 9960 ± 15740 seconds (2.8 ± 4.4 hrs) over all data sets. Figure 6 (b) shows the distribution of running time divided by input length for the three algorithms to illustrate how the running time data are distributed when normalized by length. DISCO had a slightly higher median (0.90) than FOLDALIGN (0.70), but a lower mean and standard deviation (1.46 ± 1.40 and 1.68 ± 3.35, respectively for DISCO and FOLDALIGN). These comparable numbers indicate that DISCO and FOLDALIGN are roughly equally affected by the total

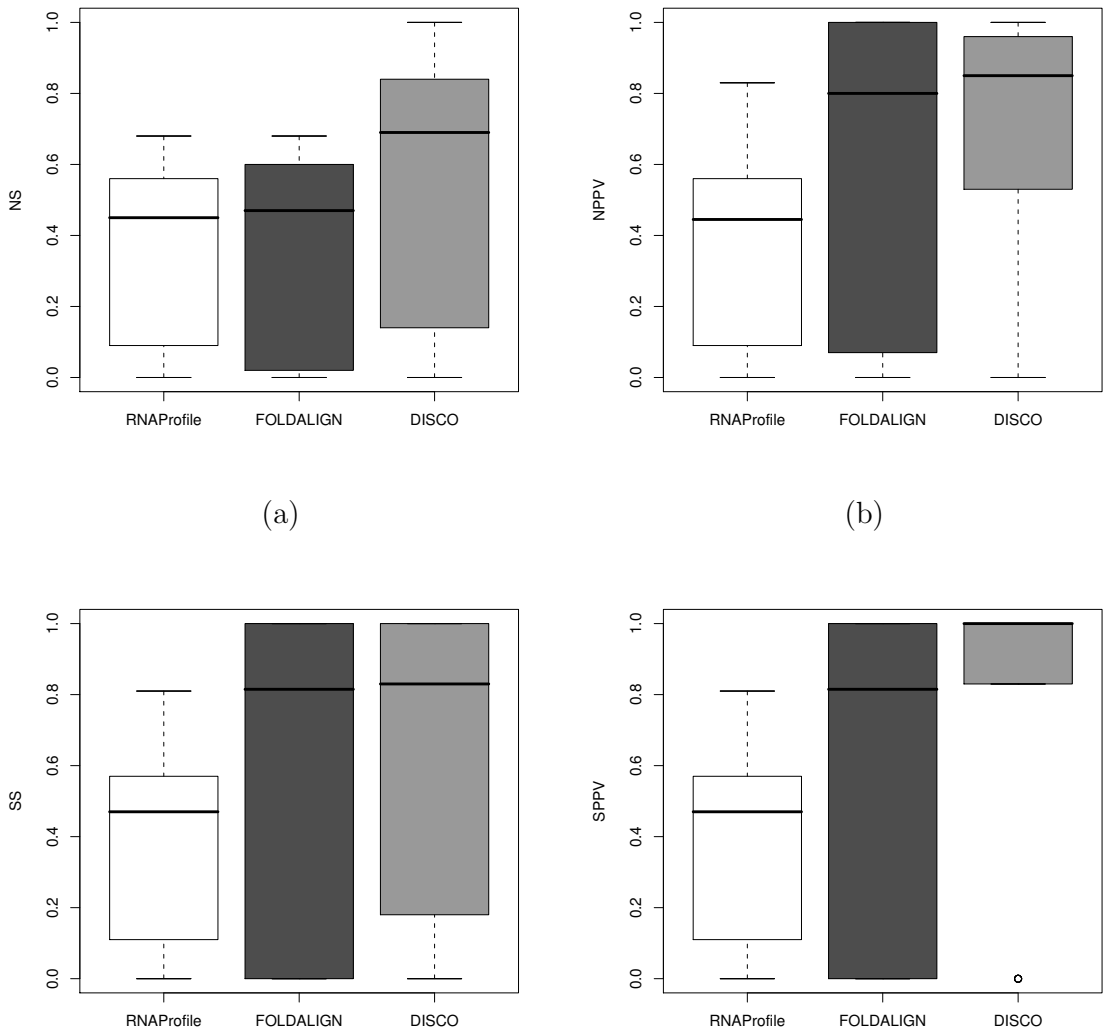
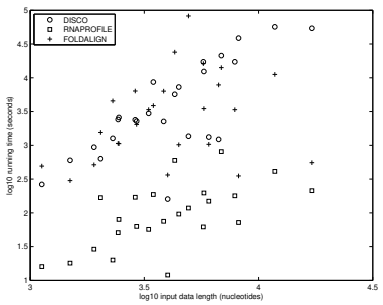
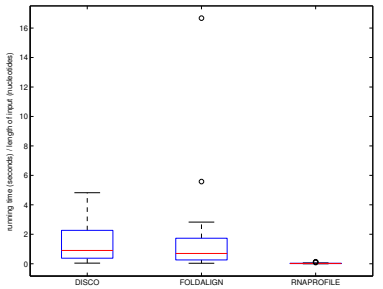


Figure 5: Box-and-whisker plots showing distributions of (a) NS, (b) NPPV, (c) SS, and (d) SPPV for RNAProfile (white), FOLDALIGN (dark grey) and DISCO (light grey).

length of the input.



(a)



(b)

Figure 6: (a) log_{10} Running time (seconds) vs log_{10} size of input data (total number of nucleotides) for DISCO, RNAProfile and FOLDALIGN. RNAProfile ran faster than DISCO and FOLDALIGN for all of these data sets. The mean log ratio of DISCO to RNAProfile was 1.6 and the mean log ratio of RNAProfile was 1.4. The mean log ratio of DISCO to FOLDALIGN was 0.1. (b) Distribution of running time (seconds) divided by length (total number of nucleotides) for DISCO, FOLDALIGN and RNAProfile. Mean $\pm$ standard deviation and median values were 1.46 $\pm$ 1.40 with median 0.90 for DISCO, 1.68 $\pm$ 3.35 with median 0.70 for FOLDALIGN and 0.03 $\pm$ 0.03 with median 0.02 for RNAProfile.

3.4 DISCO accuracy is not diminished by setting W to a general value

In our fixed parameter experiments, we set W to be equal to the width of the corresponding seed alignment of each data set. To test the effect of using a more general value for W

across all data sets, we re-ran DISCO on the miRNA data sets, setting $W = 89$ for all data sets which was the median width of the miRNA seed alignments. For comparison, we also re-ran RNAProfile using more general settings: $l = 64$ (min width of the seed alignments) and $L = 150$ (max width of the seed alignments) for all data sets. Figure 7 shows a comparison of accuracy distributions for RNAProfile runs with the specific settings (RNAP-A), the RNAProfile runs with the general settings (RNAP-B), the FOLDALIGN results on the miRNA data sets (FALIGN), the DISCO runs with the specific W settings (DISCO-A) and DISCO runs with the general W setting (DISCO-B). Table 3.4 shows the mean, standard deviation, and median values for NS, NPPV, SS, and SPPV for RNAP-A, RNAP-B, FALIGN, DISCO-A and DISCO-B. Mean values for DISCO-A and DISCO-B were higher than FALIGN, RNAP-A and RNAP-B for NS, NPPV and SPPV, although the median NPPV was higher for FALIGN. FALIGN had the highest mean value for SS. The distributions for DISCO-A and DISCO-B were very similar for NS, NPPV and SS, indicating that by these measures, DISCO is tolerant of a general setting for W. However SPPV values were lower for DISCO-B than DISCO-A, indicating that at the sequence level, some false positive predictions are introduced by the general setting for W.

	RNAP-A	**RNAP-B**	**FALIGN**	**DISCO-A**	**DISCO-B**
NS	0.43±0.19 (0.46)	0.35±0.19 (0.44)	0.57±0.23 (0.56)	0.60±0.30 (0.73)	0.61±0.28 (0.72)
NPPV	0.43±0.19 (0.49)	0.40±0.21 (0.49)	0.77±0.35 (0.98)	0.79±0.32 (0.91)	0.80±0.25 (0.90)
SS	0.45±0.19 (0.50)	0.40 ± 0.21 (0.50)	0.73 ± 0.40 (0.89)	0.69 ± 0.34 (0.83)	0.67 ± 0.40 (0.83)
SPPV	0.45±0.19 (0.50)	0.40 ± 0.21 (0.50)	0.73±0.40 (0.89)	0.87±0.33 (1.00)	0.73±0.43 (1.00)

Table 1: Mean ± standard deviation and median (parentheses) for accuracy of RNAProfile with specific parameter settings (RNAP-A), RNAProfile with general parameter settings (RNAP-B), FOLDALIGN (FALIGN), DISCO with specific parameter settings (DISCO-A) and DISCO with general parameter settings (DISCO-B). These summary statistics were computed over results for the nineteen miRNA test data sets.

4 Discussion

We developed an algorithm called DISCO to detect the most likely covariance model (CM) representing a motif embedded in a given set of unaligned RNA sequences. We tested our algorithm on 26 data sets from Rfam from two categories of RNA molecules: miRNAs and UTR elements. The data sets we constructed consisted of selected members of the Rfam family flanked by genomic or UTR sequence so that each instance of the motif was embedded in a larger sequence. Our algorithm performed quite well for the miRNA data sets and showed a type of bi-modal distribution for the UTR elements where the motif was very accurately found, or it was not found at all. We found that the measures of accuracy were not significantly correlated with any inherent properties of the input data, indicating that the algorithm has an unbiased performance with respect to sequence similarity of the motif instance, length of the motif instances, length of the input data and the number of sequences in the input data. A comparison with similar algorithms, RNAProfile and FOLDALIGN, showed that DISCO produced more sensitive results with higher positive predictive value.

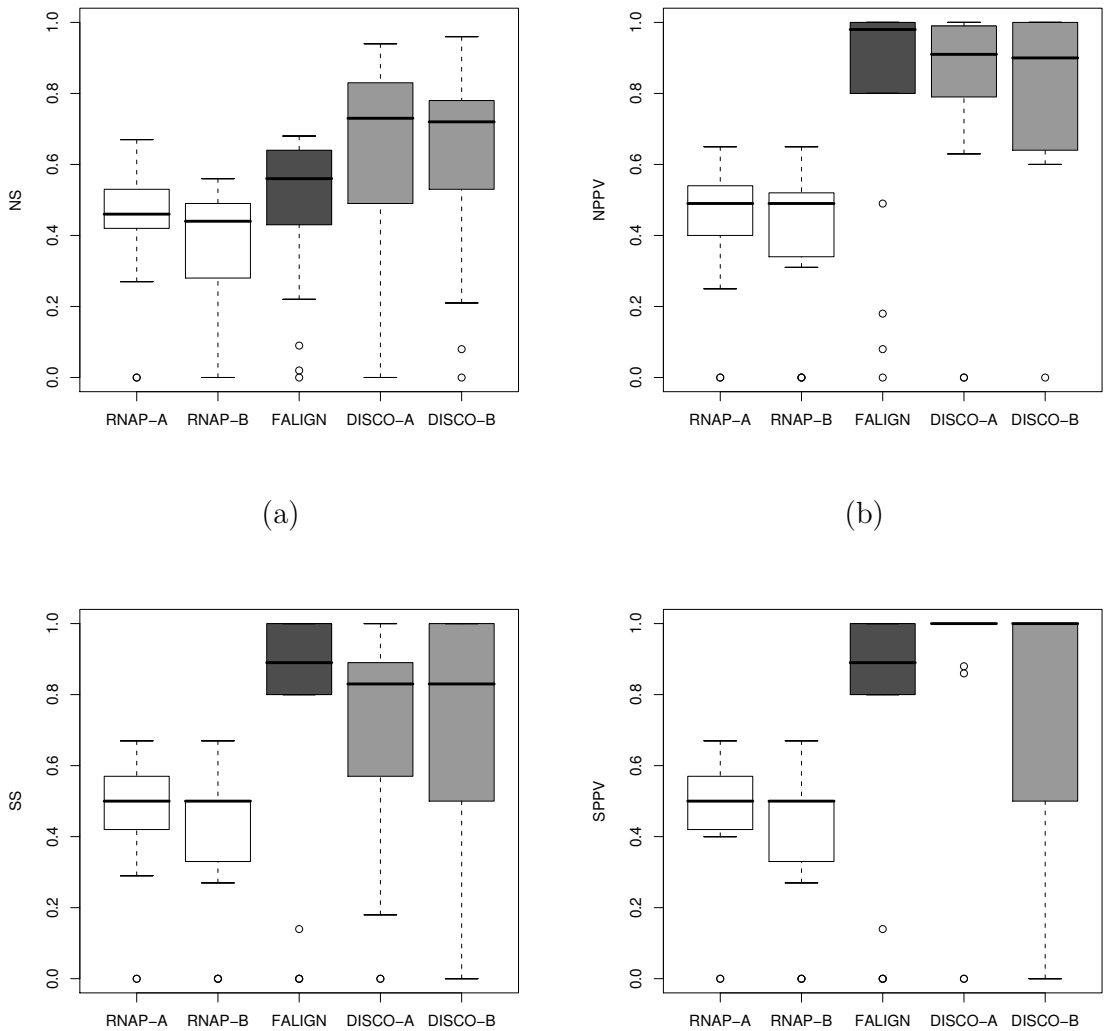


Figure 7: Comparison of accuracy RNAProfile with specific (RNAP-A) and general (RNAP-B) parameters, FOLDALIGN (FALIGN) and DISCO with specific (DISCO-A) and general (DISCO-B) parameters. Mean, standard deviation and median values are shown in Table 3.4. In general DISCO showed the best performance and with the exception of SPPV, there was no discernable difference in performance between DISCO-A and DISCO-B. This indicates a tolerance of the algorithm to a general setting of W admitting a small number of false positive predictions at the sequence level.

We found DISCO to be tolerant of a setting a key parameter to a general value without

serious impact on accuracy. Finally, we determined a positive correlation between the score output by DISCO and two sensitivity measures of accuracy.

4.1 Sequence information is more important than secondary structure in the initialisation phase

Our results indicated that the sequence method of alignment was far superior to the structure and the combination methods (see Figure 1). These results show that the sequence carries more information than the secondary structure and that sequence information is generally sufficient to create a crude multiple alignment to initialise a CM, which necessarily introduces secondary structure information in the refinement phase.

Surprisingly, there was no statistically significant correlation between accuracy and pairwise sequence identity of the motif sequences using the sequence method. This is counter-intuitive and merits further study.

For the RF00185 test data set in the fixed parameter experiments, the accuracy was 0 for all methods. However, we re-ran the algorithm with the same parameters except we used the structure alignment method instead. The accuracy results were $NS = 1.00$, $NPPV = 0.88$, $SS = 1.00$, and $SPPV = 1.00$. Although not as extreme, a similar improvement in accuracy using the structure method was achieved for RF00180, where the sequence results for all measures were 0, but the structure method gave: $NS = 0.54$, $NPPV = 0.48$, $SS = 1.00$ and $SPPV = 1.00$. Score results were 322 and 578 for the sequence and structure method respectively. These two examples indicate that, while the sequence method of alignment gave the most accurate results in general, the structure method is superior for specific data sets. More work is needed to see if there are detectable properties in the data that could suggest an appropriate choice of the sequence or structure alignment method.

4.2 Relatively few sequences can be used to initialise the CM

We ran our algorithm on data sets in which the number of sequences, N, ranged between four and nineteen sequences. The number k of aligned subsequences from which the CM was constructed was set to 6 for all data sets with $N \geq 6$ and was set to N otherwise. There was no statistically significant bias detected when the accuracy measures were tested for correlation with N (tested with Pearson product moment correlation test - data not shown). This indicates that in general, the algorithm can be successful in creating an initial CM when k is fixed independent of N.

4.3 The unpaired nucleotide filter improves performance but does not compromise accuracy for miRNA data sets

When designing the algorithm we were concerned with the $O(N^2L^2W^2)$ term of the run-time complexity. Recall that this term arises from the exhaustive pairwise alignment of all W-mers in the input. For large data sets, this step is very expensive, so we introduced a threshold measure to reduce the number of pairwise alignments performed. Only W-mers

with a proportion of unpaired nucleotides lower than a user inputted d were considered for pairwise alignment. Of major concern was whether this threshold eliminated W-mers that were motif instances in the data. For the fixed parameter experiments, we used $d = 0.55$. We repeated the experiments with $d = 0.40$ for the miRNA data, known to be highly structured, and the results were satisfactory for most data sets (see Figure 6 in Supplemental Material). Mean NS for $d = 0.40$ was 0.52 ± 0.32 with median 0.68 compared to the $d = 0.55$ results, where mean NS was 0.60 ± 0.30 with median 0.72. NPPV for $d = 0.40$ was 0.66 ± 0.36 with median 0.85 compared to the $d = 0.55$ results, where mean NPPV was 0.79 ± 0.32 with median 0.91.

We view it as a strength of our system that it can be tuned to take advantage of the structural properties of the motif if they are known ahead of time. Recent work by Bonnet *et al.* [4] and Washeitl *et al.* [40] has shown that minimum free energy signals are detectable in certain types of RNAs and our algorithm is poised to take advantage of this information.

Some motifs, however, are highly unstructured, and would not be detectable with only a minimum threshold. Implementing a maximum threshold as well would provide a range of proportion of unpaired nucleotides for W-mers to be admitted into the search space. We believe this idea should be explored further and would further enhance our algorithm.

4.4 Poor UTR element results

The algorithm completely missed the motif in four out of nine UTR data sets. Given that three of these four sets were UTRs in predominantly mammalian mRNAs, it is not too surprising that the 'sequence' alignment method for the initialisation phase presented non-motif sequences to the refinement phase: Considering the proximity of the UTRs to coding sequence, it is reasonable to assume that these sequences may be under evolutionary selection pressures to maintain their sequence. Shabalina *et al.* [38] recently reported the existence of highly conserved sequences in UTRs, detected through a genome wide comparison of orthologous mRNAs from eukaryotic species. Highly conserved patterns at the sequence level would most certainly influence the performance of our algorithm, which is not specifically designed for mRNAs. Pedersen *et al.* [35] introduce a comparative method for finding and folding RNA secondary structures within protein-coding regions. This work is of specific interest to the problem of detecting UTR elements and should be carefully considered in any modifications to our work that deal with biases in mRNA sequences.

4.5 Comparison of DISCO, RNAProfile and, FOLDALIGN

Our algorithm shows better accuracy than RNAProfile and FOLDALIGN (see Figure 5) yet is considerably slower that RNAProfile (see Figure 6). We attribute both the superior accuracy and slower running time to the use of CMs. The Needleman-Wunsch based alignment algorithm of RNAProfile considers each position of the sequences to be independently derived. One strength of the CM INSIDE algorithm is the use of transition probabilities between the states in the CM data structure which correspond to basepairs or unpaired bases in the sequence. The transition probabilities introduce a relationship between these states,

which potentially confer an advantage over the profiles and associated alignment algorithm used by RNAProfile.

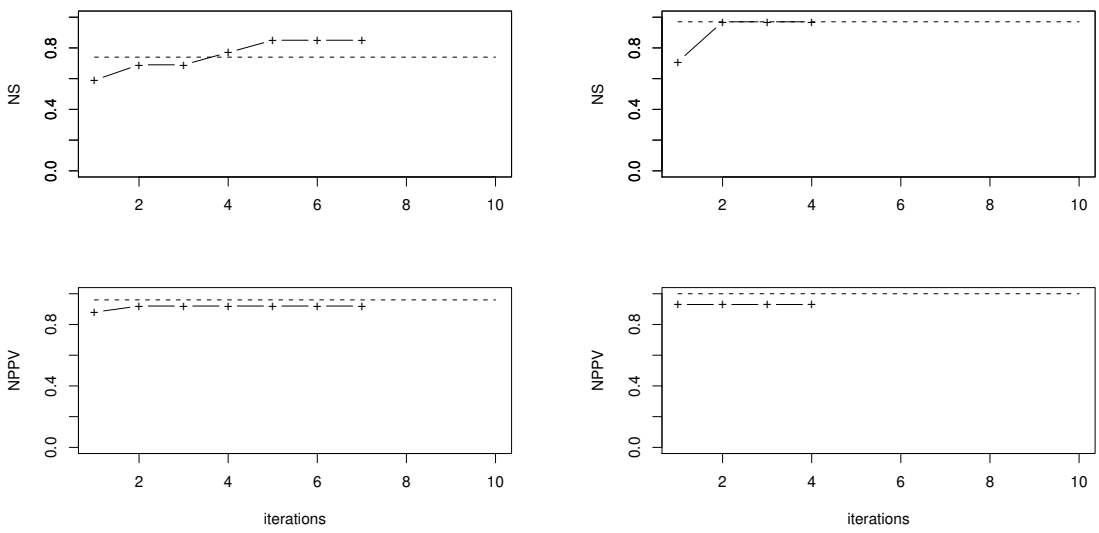
While we were expecting DISCO to be considerably slower than RNAProfile, we were not expecting it to be marginally slower than FOLDALIGN. To assess potential improvement to the DISCO running time, we looked at the l alignments (see Section 2.1.3) that were passed to the iterative refinement phase for each of the 26 test data sets. We determined the rank out of $l = 15$ of the original alignments for which, after iterative refinement, the best scoring CM was output. The mean rank was 4.8 ± 4.55 and the median was 2.5 indicating that we were on the conservative side in setting $l = 15$. In practice, reducing l will result in a reduction in running time proportional to the reduction in l. We also note that for the bulk of the data sets, iterative refinement converged in six iterations or less. Therefore an additional running time savings should be possible by setting $T = 6$ instead of $T = 10$ without loss of accuracy.

4.6 Improvements on other methods

We introduced three improvements on other methods in our algorithm. First, we used the powerful probabilistic framework offered by CMs both to model and detect motifs in our input data. With the exception of SLASH [12], none of the other methods described in Section 1 model motifs in this way. The use of CMs have a great advantage in that they offer a sensitive alignment algorithm (INSIDE) to search for an instance of a CM in a given sequence. With respect to our work, this has a two-fold benefit in that the INSIDE algorithm can be used in the refinement phase and that the output of DISCO can be easily used to detect instances of the predicted motif in other sequence databases of interest. Second, we introduce iterative refinement to the RNA motif discovery problem. None of the methods described in Section 1 use iterative refinement. Given its widespread use in the sequence motif finding domain, we believe the use of this technique is worthwhile, and confers an advantage to our algorithm. Figure 8 shows four examples that track accuracy over iterations. For the 17 miRNA data sets, there was only one case (Figure 8 (d)) where iterative refinements made accuracy markedly worse. In three other cases (including the example shown in Figure 8 (c), there was a slight, but negligible decrease in accuracy due to iterative refinement. In all other cases, accuracy was either improved or maintained by iterative refinement. Third, while DISCO with our default parameter settings is slightly slower than FOLDALIGN, our complexity bounds indicate that DISCO's initialization phase should be faster than that of FOLDALIGN, and our empirical analysis shows that significant further speedup could be obtained by reducing number of initialization models (l) and the number of iterative refinements (T).

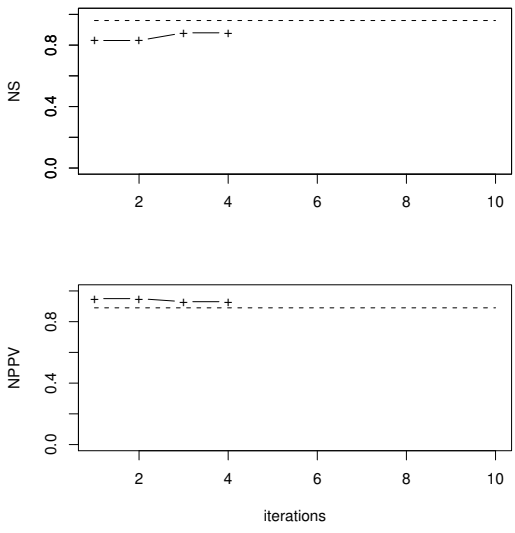
4.7 Drawbacks and limitations of the DISCO method

We acknowledge several drawbacks to our approach. Perhaps the most significant limitation is the need to specify the approximate width W of the motif. Tools such as CARNAC [39] and RNAProfile [34] require different input parameters. RNAProfile only requires the

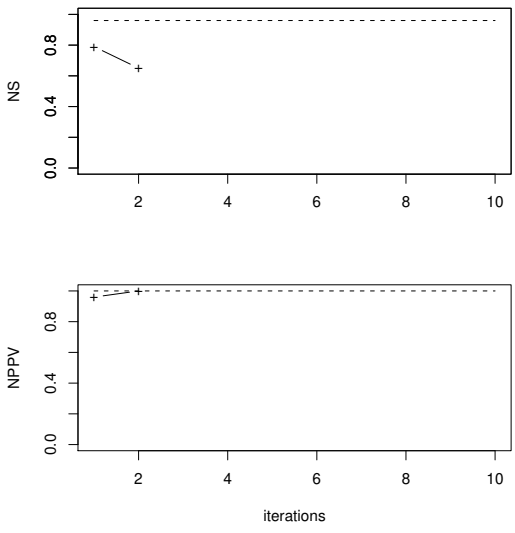


(a) RF00027

(b) RF00256



(c) RF00052



(d) RF00103

Figure 8: Accuracy plots for four miRNA data sets showing how accuracy (NS - top and NPPV - bottom) changed during the iterative refinement phase. The dashed horizontal line on each plot indicates the accuracy level of DISCO where the refinement phase was initialised with the CM of the Rfam seed alignment instead of the models computed during the initialisation phase. This line represents the best possible initialisation.

number of stems the motif is expected to have. CARNAC does not require any other input except the unaligned sequences. The need to specify W is a limitation, but it should be noted that the sequences that make up the outputted CM need not be exactly W nucleotides long.

Recall that the INSIDE algorithm allows for insertions and deletions and so the multiple alignment used in the refinement phase of the algorithm is expected to contain variable-length sequences. Furthermore, we demonstrated the algorithm is still effective on miRNA data if a more general value for W is used (see Figure 7).

4.8 Limitations of covariance models

While providing a robust probabilistic framework for modeling sets of related RNA sequences, CMs have two notable limitations. First, the bifurcating tree structure underlying the CM is incapable of modeling pseudoknots. Our algorithm will not be able to detect pseudoknots which are detectable with comRNA [20]. Second, the complexity of the INSIDE algorithm is $O(L^3)$ where L is the length of the sequence. This makes our algorithm prohibitively expensive to run on long sequences. However, Weinberg and Ruzzo [41] recently reported a method that can filter out sequences in a database to be searched with a CM in $O(L^2)$ time with no reduction in accuracy. Use of this method in the refinement phase should be considered as a potential optimisation in addition to the parameter based solutions suggested in section 4.5.

4.9 Potential improvements and future work

While our results were encouraging, there are several areas where the DISCO algorithm could be improved.

Alignment method of initialization phase. In the initialisation phase, we tested three alignment methods. The 'sequence' alignment method was superior. For the 'structure' and 'combination' methods, we constructed a scoring matrix using intuition rather than empirical results. A rigourously derived scoring matrix for the 'structure' method would provide a more accurate comparison to the 'sequence' method which used a matrix, RIBOSUM85-60 that was derived using maximum likelihood methods under the BLOSUM model of evolution (see [21]). Given that for some data sets, the 'structure' method did outperform the 'sequence' method, we feel this further work has merit.

Use of priors when initialising the CM. A uniform Dirichlet prior was used to intialise both the transition and emission probabilities of the CM. The effect of different priors and the use of any other empirically derived statistics in the construction of the CM were not investigated. Given the numerous (more than 500) CMs now available in Rfam, it would be interesting to estimate a more data-driven prior from these existing sets. Furthermore, priors for specific types of RNAs (eg miRNAs) could be estimated and optionally used if the user had prior knowledge of the type of motif they were expecting to discover.

Use of phylogenetic weighting. In the field of comparative genomics, a growing body of literature is reporting different models to incorporate phylogenetic distance in analysing sets of sequences where the individual sequences originate from different organisms. Knudsen

and Hein [23] infer a phylogenetic tree using maximum likelihood methods and use the distances in the tree to help infer a consensus secondary structure using SCFGs. Weighting the alignment scores in the initialisation phase could more accurately reflect the similarity of the sequences and, in effect, normalise the scores by evolutionary distance. The work of Holmes [18] describes an evolutionary model for RNA structure and its use in constructing pair-SCFGs to align two homologous RNAs. Exploring the use of such evolutionary models for RNA sequences could improve the accuracy of motif detection in sequences from different organisms.

5 Acknowledgements

We thank Dr. Wyeth Wasserman from the Department of Medical Genetics, University of British Columbia (UBC) for helpful comments and guidance throughout the course of this project. We thank Mirela Andronescu from the Department of Computer Science, UBC for providing code to compute the minimum free energy secondary structure prediction. SPS and AC were both supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

[1] M. Andronescu, ZC. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *J Mol Biol*, 345(5):987–1001, Feb 2005.

[2] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximizationm. *Machine Learning*, 21:51–83, 1995.

[3] T.L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol*, 2:28–36, 1994.

[4] E. Bonnet, J. Wuyts, P. Rouzé, and Y. Van de Peer. Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics*, 20(17):2911–2917, Nov 2004.

[5] J. Buhler and M. Tompa. Finding motifs using random projections. *J Comput Biol*, 9:225–242, 2002.

[6] JL. Casey, MW. Hentze, DM. Koeller, SW. Caughman, TA. Rouault, RD. Klausner, and JB. Harford. Iron-responsive elements: regulatory RNA sequences that control mRNA levels and translation. *Science*, 240(4854):924–928, May 1988.

[7] R. Durbin, S.R. Eddy, Krogh A., and Mitchison G. *Biological sequence analysis*. Cambridge University Press, 1998.

[8] SR. Eddy. Non-coding RNA genes and the modern RNA world. *Nat Rev Genet*, 2(12):919–929, Dec 2001.

[9] SR. Eddy. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, 3(1):18–18, Jul 2002.

[10] SR. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Res*, 22(11):2079–2088, Jun 1994.

[11] J. Gorodkin, LJ. Heyer, and GD. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res*, 25(18):3724–3732, Sep 1997.

[12] J. Gorodkin, SL. Stricklin, and GD. Stormo. Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res*, 29(10):2135–2144, May 2001.

[13] JH. Havgaard, R. Lyngso, GD. Stormo, and J. Gorodkin. Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*, Jan 2005.

[14] CU. Hellen and P. Sarnow. Internal ribosome entry sites in eukaryotic mRNA molecules. *Genes Dev*, 15(13):1593–1612, Jul 2001.

[15] MW. Hentze and LC. Kühn. Molecular control of vertebrate iron metabolism: mRNA-based regulatory circuits operated by iron, nitric oxide, and oxidative stress. *Proc Natl Acad Sci U S A*, 93(16):8175–8182, Aug 1996.

[16] IL. Hofacker, M. Fekete, C. Flamm, MA. Huynen, S. Rauscher, PE. Stolorz, and PF. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucleic Acids Res*, 26(16):3825–3836, Aug 1998.

[17] IL. Hofacker, M. Fekete, and PF. Stadler. Secondary structure prediction for aligned RNA sequences. *J Mol Biol*, 319(5):1059–1066, Jun 2002.

[18] I. Holmes. A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*, 5(1):166–166, Oct 2004.

[19] YJ. Hu. Prediction of consensus structural motifs in a family of coregulated RNA sequences. *Nucleic Acids Res*, 30(17):3886–3893, Sep 2002.

[20] Y. Ji, X. Xu, and GD. Stormo. A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, 20(10):1591–1602, Jul 2004.

[21] RJ. Klein and SR. Eddy. RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinformatics*, 4(1):44–44, Sep 2003.

[22] B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6):446–454, Jun 1999.

[23] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, 31(13):3423–3428, Jul 2003.

[24] M. Lagos-Quintana, R. Rauhut, W. Lendeckel, and T. Tuschl. Identification of novel genes coding for small expressed RNAs. *Science*, 294(5543):853–858, Oct 2001.

[25] A. Lambert, A. Lescure, and D. Gautheret. A survey of metazoan selenocysteine insertion sequences. *Biochimie*, 84(9):953–959, Sep 2002.

[26] NC. Lau, LP. Lim, EG. Weinstein, and DP. Bartel. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science*, 294(5543):858–862, Oct 2001.

[27] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[28] C.E. Lawrence and A.A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1990.

[29] TM. Lowe and SR. Eddy. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*, 25(5):955–964, Mar 1997.

[30] TM. Lowe and SR. Eddy. A computational screen for methylation guide snoRNAs in yeast. *Science*, 283(5405):1168–1171, Feb 1999.

[31] TJ. Macke, DJ. Ecker, RR. Gutell, D. Gautheret, DA. Case, and R. Sampath. RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res*, 29(22):4724–4735, Nov 2001.

[32] BW. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta*, 405(2):442–451, Oct 1975.

[33] JS. Mattick. Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Rep*, 2(11):986–991, Nov 2001.

[34] G. Pavesi, G. Mauri, M. Stefani, and G. Pesole. RNAProfile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences. *Nucleic Acids Res*, 32(10):3258–3269, 2004.

[35] JS. Pedersen, IM. Meyer, R. Forsberg, P. Simmonds, and J. Hein. A comparative method for finding and folding RNA secondary structures within protein-coding regions. *Nucleic Acids Res*, 32(16):4925–4936, 2004.

[36] E. Rivas and SR. Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7):583–605, Jul 2000.

[37] E. Rivas, RJ. Klein, TA. Jones, and SR. Eddy. Computational identification of noncoding RNAs in E. coli by comparative genomics. *Curr Biol*, 11(17):1369–1373, Sep 2001.

[38] SA. Shabalina, AY. Ogurtsov, IB. Rogozin, EV. Koonin, and DJ. Lipman. Comparative analysis of orthologous eukaryotic mRNAs: potential hidden functional signals. *Nucleic Acids Res*, 32(5):1774–1782, 2004.

[39] H. Touzet and O. Perriquet. CARNAC: folding families of related RNAs. *Nucleic Acids Res*, 32(Web Server issue):142–145, Jul 2004.

[40] S. Washietl, IL. Hofacker, and PF. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc Natl Acad Sci U S A*, 102(7):2454–2459, Feb 2005.

[41] Z. Weinberg and WL. Ruzzo. Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, 20 Suppl 1:334–334, Aug 2004.

[42] Z Yao, Z Weinberg, and W L Ruzzo. Cmfinder–a covariance model based rna motif finding algorithm. *Bioinformatics*, 22(4):445–452, Feb 2006.